

LINUX EXPERT

FAQ

Q The Linux manual pages are all well and good, but can you get a brief reminder of what each command does on the command line?

A The 'whatis' command is probably what you need. Although it doesn't give any clue as to the arguments any particular command accepts, it searches the manual pages and provides a brief summary. More importantly, it also provides a reference to the relevant manual pages.

```
# whatis vi
vi (1) - Vi Improved, a
  programmer's text editor
vi (ip)- screen oriented
  (visual) display editor
```

For instance, the above example tells us that you can get more information about the vi command by viewing the manual pages with the reference in brackets, like this:

```
# man 1 vi
# man ip vi
```

If you come across a command you don't know, rather than run it to see what happens, it's safer to use **whatis**. To see which manual pages tell you about the **whatis** command itself, you can use the following command:

```
# whatis whatis
```

Jon Thompson
UNIX/Linux management
and security consultant



linux@computershopper.co.uk

Using MySQL databases

A professional database can cost hundreds of pounds, but there is a fast, powerful and free alternative. **Jon Thompson** explains how to set up and use the popular MySQL package

MySQL is a flexible and easy-to-use database that is suitable for a wide range of purposes, from organising a basic contacts list to running a massive, multi-regional, fault-tolerant information system. It is often used to power dynamic websites in combination with a programming language such as PHP, and many web-based email services use MySQL to store messages. It's free, too, so you can use it to run small personal projects and gain skills that could be useful in a larger, commercial situation.

MySQL is fast, reliable and can handle multi-terabyte databases. The code is multi-threaded for efficiency and scalability, and it can use multiple CPUs if available. It can also replicate the data it holds across several slave servers for fault tolerance, and can handle its own backup and restoration duties. As well as being an industrial-strength database server, MySQL has a suite of associated graphical tools that enable you to create and manage databases.

We'll show you how to set up MySQL on a Linux system, how to check that it is running properly, run queries and back up your data. Before you start you'll need to decide which version of MySQL to use.

DIFFERENT STROKES

The version you will probably want to use is the free one, called MySQL Community Server. It may be available on your system already, or on your original installation CD set. If not, you can download it from www.mysql.org/downloads (see Installation, below). Although this code is free to download and use, MySQL also comes in various commercial packages, comprising software, services and support. For corporate users, there's MySQL Enterprise. This comes with a bug fix service to keep you up to date, and also the Network Monitoring and Advisory Service. This piece of software actively monitors the database server and advises on actions to take when it senses developing problems.

If you have mission-critical applications that must be available no matter what, MySQL Cluster is a good choice. This is designed to remain running and, according to the website, can provide 99.999 per cent availability. There is also MySQL Cluster Carrier Grade. This is designed for communications companies that need to be able to design flexible, fault-tolerant databases spread across various geographical locations.

MySQL Embedded reflects the increasing need for consumer devices to have access to their own databases. The developers also provide a range of drivers to enable seamless integration over ODBC, .NET and a host of other languages and connection methods.

PLAYS WELL WITH OTHERS

A number of computer languages that interface with MySQL are available on Linux. In fact some languages, such as PHP, are designed primarily to process and shunt data between web servers and back-end databases. Any website you visit that requires you to register will use some form of database, and increasingly, it's MySQL.

MySQL understands an interpreted language called Structured Query Language (SQL). This plain-text language is what the client end of an application sends to the database server. SQL's vocabulary and syntax is very rich and enables the database programmer to express very complex ideas and operations. Although as with any language there are several dialects, there's always the standardised ANSI SQL, which MySQL can also be made to accept rather than its own version.

What people tend to think of as a database is actually the server-side program that takes SQL queries and processes them against the files in the databases it manages. The server is more properly called a database management system (DBMS). Like Oracle and SQL Server, the databases MySQL manages are relational.

A relational database contains tables that consist of columns and rows. Each row stores a record made up of columns that carry data of a predetermined type: a name, a number, a date and so on. As you enter data, each table grows. A relational database management system enables you to create a relationship between tables to form a complex structure of information with the minimum amount of duplication.

Say, for example, that you have an order-processing database and want to find out about a specific customer's orders. With a relational database you would select a customer number in a table containing customer details and then search a table containing the orders for all orders that have a matching customer ID number. The advantage of storing the customer's details in one table and the orders in another is that you don't replicate the customer's address and so on.

The query below will display the names of the ordered items listed in the orders table where the customer number matches the one selected from the customer details table:

```
SELECT item FROM order WHERE customer = '37'
```

If you wanted to find out where this customer lives, you would run a query that looks up his address from the customer details table, like this:

```
SELECT address FROM customer WHERE id = '37'
```

INSTALLATION

Before we install MySQL and its graphical tools, consider whether you will run the database server and client on the same system. If you need them to run on different computers, you must open TCP port 3306 on the server system's firewall, because MySQL uses it for communication between client and server. It's also worth noting that, for security reasons, you should never set up a database server so that it faces the internet directly. If it needs to communicate with a web server, this can happen in a demilitarised zone between the internet and your internal network. Never let any traffic flow directly between the internet and your internal network, and never let people on the internet connect directly to your database.

Installing MySQL, like all the mainstream Linux server packages, is simplicity itself using a package manager, because the server should be bundled with your distribution.

For users of SuSE and OpenSUSE, installation is easy with the YaST system configuration tool. Install the `mysql`, `mysql-administrator` and `mysql-query-browser` packages. In Ubuntu, go to the System pull-down menu, select Administration and run the Synaptic Package Manager. If the administrator and query-browser packages aren't in the list when you search for "mysql", don't worry. We'll show you how to download and install these later.

Ubuntu and Debian users who don't have a graphical installation tool still have the useful `apt-get` command-line tool at their disposal. Run these commands to install the server and tools:

```
# apt-get install mysql-server
# apt-get install mysql-gui-tools
```

Note the names of the packages to install. If you've configured your `apt-get` sources file correctly, it will download and install the latest MySQL package. Ubuntu users will find that they are not running as root by default. To run the above commands you'll have to prefix each with the 'sudo' command and enter your password:

```
# sudo apt-get install mysql-server
Password:
```

You can use the following command to become root and not have to use the `sudo` command each time:

```
# sudo su
Password:
```

Enter the password you used to log into the desktop and you should see the familiar hash prompt that characterises a root login. Then type these commands:

```
# apt-get install mysql-server
# apt-get install mysql-gui-tools
```

If MySQL isn't bundled with your distribution and you don't have access to `apt-get`, you'll have to download it and install it by hand. The MySQL team have compiled some handy RPMs for you to download and install directly, which you can find at www.mysql.org/downloads.

If you're in doubt, download and install one of the generic RPM packages, which are at the bottom of the list. These are available for both the x86 and 64-bit families of Intel microprocessors and their AMD variants.

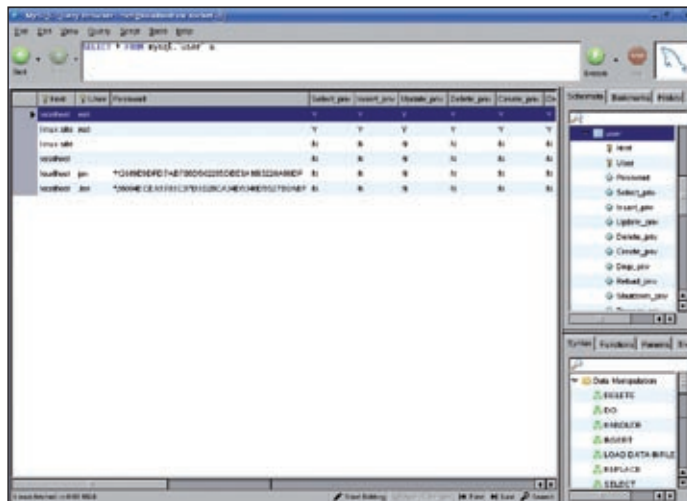
Once you have selected a package to download, your choice will link to a page showing the various downloads available. Select 'Server' and download it. You may also see a 'Max Server' option, but while this is packed with extra features, not all are as tested as they should be – as the MySQL team points out.

Once downloaded or extracted from your distribution media, you can install the MySQL RPM using the standard `rpm` command.

```
# rpm -i <rpm name>
```

SAFETY FIRST

Having installed MySQL, we must configure it to create some basic system tables. We can do this



◀ You can build and run queries in Query Builder or construct and run entire scripts consisting of SQL statements

with the following command, entered as root:

```
# mysql_install_db
```

This command creates everything MySQL needs to handle user accounts and system privileges. The output of this command includes several pieces of advice, the most important of which is to set a password on MySQL's root account. This is separate from the Linux root account but works in a similar way, in that it is used for administrative access to the MySQL server. Set the root password with the following command:

```
# mysqladmin -u root password
<password>
```

Ensure that <password> is the password you want to set. We can now test that MySQL is functioning properly. First, go to `/etc/init.d` and enter the command:

```
# ./mysql status
checking status for service MySQL:
                                running
```

If the status returned is 'stopped' or 'not used', start the server with the following command:

```
# ./mysql start
Starting service MySQL
                                done
```

With the server running, you can now try to connect to it. To do this, while logged in on the server, enter the following command:

```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.
Commands end with a ; or \g.
Your MySQL connection id is 12
to server version 5.0.38-standard
```

Type 'help' or '\h' for help. Type '\c' to clear the buffer.

```
mysql>
```

To quit the MySQL command line, issue the 'quit;' command, including the trailing semicolon. You can also see the default databases created

by the `mysql_install_db` command:

```
mysql> show databases;
```

```
+-----+
| mysql |
| test  |
| tmp   |
+-----+
3 rows in set (0.00 sec)
```

This shows three databases. You can examine the tables contained within a particular database, in this case the database called `mysql`, with the following commands:

```
mysql> use mysql;
mysql> show tables;
```

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
| user_info       |
+-----+
16 rows in set (0.00 sec)
```

You can also examine the columns that form the tables, as shown at the top of the next page.

ADMINISTRATION MADE EASY

The developers provide graphical tools to help you run, administer and query your database. Your distribution should include these in its `mysql` packages, but MySQL also provides them in the `mysql-gui-tools` package, from <http://dev.mysql.com/downloads/gui-tools/5.0.html>. ➔

```
mysql> describe user_info;#
```

Field	Type	Null	Key	Default	Extra
User	varchar(16)		PRI		
Full_name	varchar(60)	YES	MUL	NULL	
Description	varchar(255)	YES		NULL	
Email	varchar(80)	YES		NULL	
Contact_information	text	YES		NULL	
Icon	blob	YES		NULL	

6 rows in set (0.00 sec)

▲ You can examine the columns that form the tables in your MySQL database

Download the generic Linux tarball, which contains the binaries with their dependencies (libraries and so on) already compiled.

The package itself contains two very useful applications. First, there's MySQL Administrator, which is a powerful visual administration console. Administrator's great strength is that it enables you quickly to gain an understanding of how your database server software is configured and what's happening as it runs. It allows you to perform all the database administration tasks that you would usually perform laboriously on the command line.

Another useful tool is MySQL Query Browser. This enables you to create complex SQL queries graphically using a drag-and-drop interface. It can also create and run SQL scripts, and provides comprehensive help on each part of the SQL syntax.

To install the mysql-gui-tools package, run the following tar command with the '--directory' switch:

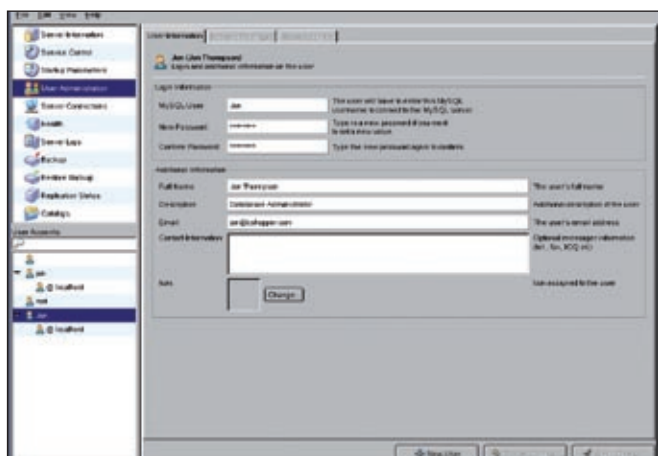
```
# tar --directory=/opt -xzf mysql-gui-tools-<version>.tar.gz#
```

The tools themselves will be installed to the /opt directory, as recommended by the MySQL website. In the directory called /opt/mysql-gui-tools-version/share/mysql-gui, you'll find desktop icons that you can copy to your desktop. Click on these to launch the various tools.

USING THE TOOLS

Start MySQL Administrator with the following command or, if you've installed it from the tarball, by clicking on its desktop icon:

```
# mysql-administrator#
```



▲ Exercise easy control over MySQL users with Administrator

You'll see the MySQL Administrator logon screen. To be able to connect to the server, you should enter the server name as 'localhost', the user as 'root' and use the password you assigned to the MySQL root account earlier. Press the connect button and you should see the main MySQL Administrator interface appear. Use the localhost server setting

only if you're logged into the same system that's running MySQL. If MySQL is running on another computer on the network, you'll need to enter its IP address or DNS name to access it remotely.

If, when you try to connect, you see a pop-up box explaining that MySQL Administrator couldn't connect to the server software, it's possible that you have either entered an incorrect password or that the server isn't running. Helpfully, you'll also be presented with a button enabling you to ping the server to make sure the machine can at least be reached.

Once you're logged in, MySQL Administrator shows the initial server status screen. This gives basic details such as the user you're logged in as, the host you're connected to and various other pieces of information about the MySQL server and the hardware on which it runs.

You should see 11 categories covering every aspect of server operation listed down the left-hand side of the screen. Click on the Service Control category. The resulting 'Stop the Server' button is one that you should avoid pressing while users are connected. Interestingly, once the server has stopped, its status as reported by the /etc/init.d/mysql script will be set to 'unused'. If you're not sure what a button does, let the mouse pointer hover over it and a tool tip will appear, either giving extra information about the option or giving the name of the configuration file parameter to which it corresponds.

The next category includes the very extensive Startup Parameters. This covers the configuration file parameters, which are split by subject into nine tabs. Unless you're into serious database tuning and replication, you'll probably never use most of these, but Replication is interesting. It's a way of keeping a running backup on a group of slave servers, which form a loose cluster and

continue to service the database if the primary server fails. There's an excellent tutorial on replication at www.howtoforge.com/mysql_database_replication.

It is worth paying some attention to the security tab towards the right. This includes a setting that deactivates the 'show database' command, limiting the amount of information users can get about the structure and content of databases. You can also make an entire database read-only on this screen, meaning that it can only ever be used for reference. If you're running a website with static information that only you can update, using read-only mode should guard against SQL injection attacks.

Another useful setting is Safe User Create. When this option is checked, users are unable to create other users with the SQL grant statement, unless they also have insert privileges for the mysql.users table.

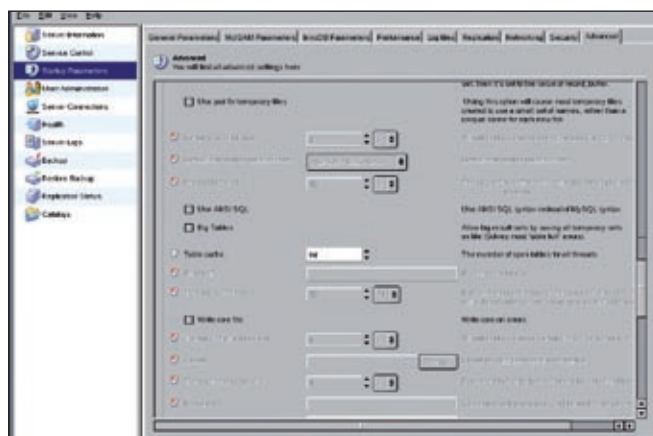
We mentioned earlier that MySQL can be made to accept only ANSI SQL. If the application you're installing speaks ANSI SQL, and it's the only one that uses this MySQL server, you can enable ANSI-only mode by clicking on the Use ANSI SQL button, which is about three-quarters of the way down the list on the Advanced tab.

NEW USERS

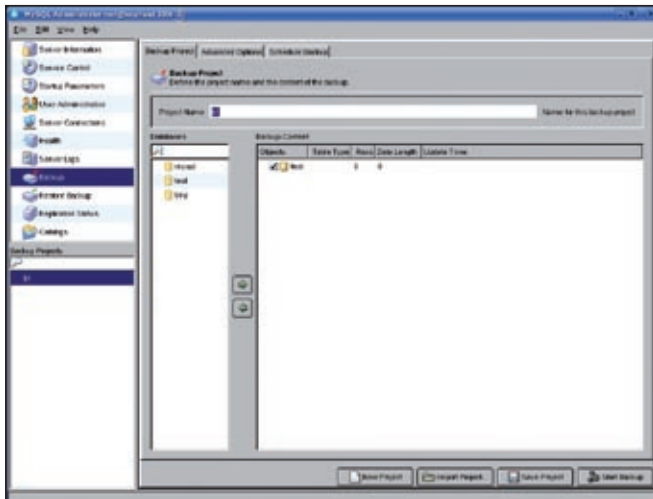
Adding a new user in MySQL Administrator is very simple. First, select the User Administration category in the left-hand pane. Note that a list of current user accounts appears below the categories, from which you can select an account to modify.

To add a user, press the New User button at the bottom of the screen. Fill in the usual details of username, password and so on, and press Apply Changes. An entry for the user will appear in the left-hand pane of the interface. Double-click on it and you'll see that the tabs for Schema Privileges and Resource Limits appear in the details pane. Click on the first, which allows you to assign privileges to the user on a database level. This in turn means you can carefully control what a user can do within that database.

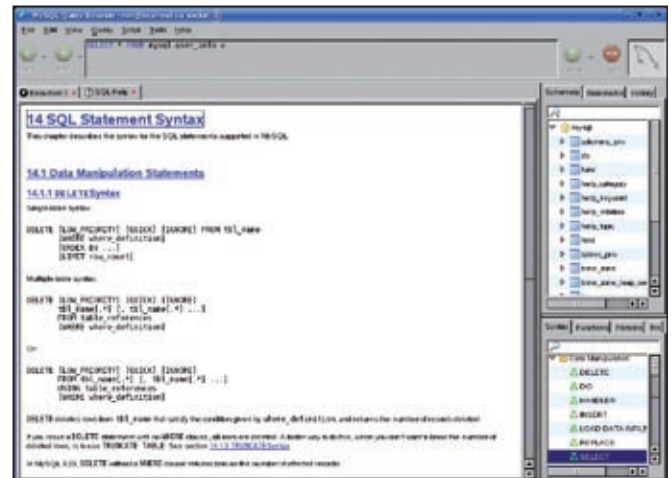
Similarly, the Resource Limits tab enables you to set limits on SQL queries, database updates and connections to the database. This is to prevent malware, a malicious user or plain bad programming eating up resources and thereby causing a denial of service attack. The defaults of zero in these fields allow unlimited use, which should be fine for regular, internal network use. You can check what's



▲ You'll find an option buried among the startup parameters that forces MySQL to accept only the standardised ANSI dialect of SQL



▲ With MySQL Administrator, you need never miss another backup



▲ As well as providing a graphical query tool, Query Builder also tells you about SQL syntax

going on with the server's connections using the Server Connections category.

MySQL is a multi-threaded server that opens up a new thread for each connection. The first tabbed window in this category shows each of these threads, as well as details of machines that have connected to databases, the database to which each is connected, the last SQL command that has been run and various other pieces of information. The second tabbed window summarises the thread information according to user. Select one and the lower half of the screen shows the threads assigned to that user.

STAY HEALTHY

The Health category is a great way of checking that the MySQL server software is up and running and that it's not overloaded. The default Connection Health tab provides a real-time display of the number of connections, the traffic flow and the number of SQL queries generating that traffic. You can add your own graphs (bar or line) or delete one from this display by right-clicking on an existing one. Creating custom graphs is a little too in-depth for this month's *Linux Expert*, but there's a good page on creating them at <http://tinyurl.com/3cthw1>.

Allied to the Health category is Server Logs, which is immediately below it. This is the first port of call if you have a problem with a server and need extra information. This category has tabs for the error log, the general log and the innodb log. This last log is for MySQL's transaction storage engine and can help diagnose serious multi-user database problems.

System administrators, especially database administrators, are responsible for backing up their servers. MySQL Administrator makes this easy by providing tools for creating and scheduling backup tasks. Select the Backup category, then press the resulting New Project button and enter a name in the Project Name field. Now select the databases you need to include in the backup. Save the project and you'll see it appear on the list of backup projects on the left of the screen under the categories.

To start the current backup project immediately, press the Start Backup button. A dialog box will open asking you where you want to place the backup file. The output is a representation of the database in SQL format, which you can then compress and store.

You can also automate your backups with the Schedule Backup tab. To use this, select the

pull-down menu and choose between a daily, weekly or monthly backup. Now select the time at which the backup should occur, the target directory and any other options and press the Schedule Backup button. To cancel a previously created schedule for this backup project, simply press the Uninstall Schedule button and the job will be removed from the system.

To make restoring your backups painless, there's the Restore Backup category. Just select the backup project you want from the list beneath the categories pane and press the Restore Backup button. That's all there is to it.

If you're using replication to keep a live copy of your databases on another server, the Replication Status category is for you. This shows the hostnames of the slaves and their statuses. Replication settings are entered in the Startup Parameters category.

The final category is Catalogs. This provides details of the tables and indices that make up the databases stored on the server. Selecting this category presents the details of the databases stored on the server. These appear under the categories pane. Select one to display the tables it contains.

Selecting a table from the list allows you to see its status and statistics, but you can also add new tables to a database and edit existing ones. Click on the Create Table button and press the right mouse button to add columns, just as you would in Microsoft Access or one of its clones. When you have finished, press Apply changes and Administrator will show you the SQL it intends to run to create the table. Confirm this, and your new table is added to the database.

JUST BROWSING

Another very useful and versatile tool is the MySQL Query Browser. This enables both SQL beginners and experienced programmers alike to construct sophisticated queries. Start it from the desktop icon you installed or use the following command line:

```
# mysql-query-browser &
```

You'll see the familiar connection dialog box. The main window of the query browser is divided into several panes. At the top is the query you're constructing. To the right, you'll see list of databases. If you select one and click on the arrow that appears next to it, you'll see a list of the available tables it contains. Click

on a table to open it, and you'll see the columns that make it up.

You can type your query into the query pane directly, or you can drag and drop the column or table you need and let the software construct a query for you. To construct a very basic query, pick a table such as 'user', which is found in the 'mysql' database, and drag one of the columns it contains to the query pane at the top of the screen. As you cross the threshold into the query pane, you'll notice that several SQL statements pop up. If you drop your table on one of these statements, a syntactically correct query that uses the statement will appear in the query pane.

If, however, you keep going and drop the table into the pane itself, the query it constructs will begin with the SELECT statement by default, and you'll interrogate the database rather than changing anything. To run the query, simply press the big green Execute button and a tabbed results pane will open up displaying the rows returned by the query.

But what do you do if you don't know any SQL? This is where Query Browser can help as a learning tool. At the bottom-right of the screen is another pane with several tabs. Each contains an area of SQL functionality, such as data manipulation statements. Click on the arrow that accompanies each entry in each tab and you'll see statements contained within that category.

For example, double-click on the DELETE entry under data manipulation. After a second or so, a tabbed help pane appears with full details on how to use the delete statement, including the arguments it takes.

You can also create longer scripts that combine individual SQL statements, which you can then run on demand. To do this, go to the File menu and select New Script Tab. You can now enter statements and save them to a file, execute the script or step through it one statement at a time. The results are displayed in a new results pane.

MySQL, like many other Linux servers, has come a long way in a short period of time. Being open source it has had the twin advantages of many people checking and improving the code continuously, and the growth of a large and very helpful user community. If you have problems with your installation of this or any other piece of Linux software, you can bet that someone else has had the same problem and, more importantly, has received an answer. You can often find such answers on Google's Groups. **CS**